

Small codes for magic state distillation

Mark Howard^{1,2,a} and Hillary Dawkins³

¹ Department of Physics and Astronomy, University of Sheffield, Sheffield, UK

² Institute for Quantum Computing and Department of Applied Mathematics, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1

³ Institute for Quantum Computing and Department of Physics and Astronomy, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1

Received 30 November 2015 / Received in final form 1st February 2016

Published online 15 March 2016

© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract. Magic state distillation is a critical component in leading proposals for fault-tolerant quantum computation. Relatively little is known, however, about how to construct a magic state distillation routine or, more specifically, which stabilizer codes are suitable for the task. While transversality of a non-Clifford gate within a code often leads to efficient distillation routines, it appears to not be a necessary condition. Here we have examined a number of small stabilizer codes and highlight a handful of which displaying interesting, albeit inefficient, distillation behaviour. Many of these distill noisy states right up to the boundary of the known undistillable region, while some distill toward non-stabilizer states that have not previously been considered.

1 Introduction

Most efforts towards building a large-scale quantum computer use error-correcting codes to protect the quantum information. However, no matter what code is chosen, the set of gates that are transversal (i.e. manifestly fault-tolerant) will be non-universal [1], meaning these operations are insufficient for useful quantum computation. Therefore, some additional resource will be required to supplement the limited set of operations. The most promising technique for circumventing this issue is to supplement the non-universal gate set with a supply of special resource states, known as magic states. Having access to pure magic states enables implementation of an additional unitary operation, which ultimately provides us with a universal gate set. The magic state distillation protocol, as introduced by Knill [2] and Bravyi and Kitaev [3], provides a method of producing these resource states through an iterative procedure in which less pure magic states are consumed to produce a higher purity magic state using only stabilizer operations.

Here we use small stabilizer codes, which means that the number of qubits, the number of measurements and the number of nonlocal operations involved in each round of distillation are all small. While these are attractive features, it seems that using a small code limits the amount of purification that can occur per round of distillation. For practical purposes, one typically quantifies how many raw magic states are required to produce a single magic state

of very high purity (e.g., infidelity of no more than 10^{-10}). Applying this metric one typically finds that small codes fare much worse than larger codes like the 15-qubit Reed-Muller code [3]. If instead we rank codes by their range of applicability, i.e., the threshold noise rate above which purification no longer occurs, then smaller codes seem to fare at least as well as larger codes. Although this may not be a pressing issue experimentally, since it seems reasonable to expect raw state preparation infidelity on the order of 10^{-2} – 10^{-4} , it is of foundational interest. Bravyi and Kitaev highlighted the fact that tight magic state distillation routines imply a sharp transition, as noise is decreased, from a circuit that is classically efficiently simulable to one that enables universal quantum computation. The question of tight distillability for qutrit states was addressed in references [4,5].

Efficient codes for magic state distillation typically exhibit quadratic ($p \mapsto \mathcal{O}(p^2)$) or cubic ($p \mapsto \mathcal{O}(p^3)$) suppression of the error parameter p . The codes listed here all exhibit linear error suppression and consequently are not competitive with existing routines in terms of efficiency. Nevertheless we feel that exploring the landscape of codes that achieve distillation is still worthwhile. Reichardt [6] has previously summarized a handful of known distillation routines exhibiting the best thresholds.

A number of the codes presented below achieve tight distillation right up to the boundary of the stabilizer octahedron – the convex hull of Pauli eigenstates depicted in Figure 1 – whose interior contains states that are provably undistillable. We also present codes that converge, upon iteration, to states that are not those that are typically

^a e-mail: m.howard@sheffield.ac.uk

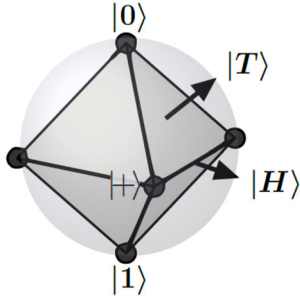


Fig. 1. The stabilizer octahedron inscribed within the Bloch sphere: Six Pauli eigenstates form the vertices of an octahedron. States within the octahedron are provably undistillable so the best one can hope for is to distill states up to the boundary. The two pure non-stabilizer states singled out here, $|H\rangle$ and $|T\rangle$, were shown to be distillable by Bravyi and Kitaev [3].

considered i.e., the H -type and T -type magic states. None of our codes require twirling (which diagonalizes the state in the $\{|H\rangle, |H^\perp\rangle\}$ or $\{|T\rangle, |T^\perp\rangle\}$ basis) between rounds.

2 Background

Here we briefly summarize notation and terminology associated with both magic state distillation and with the type of quantum codes that we have used.

2.1 Magic state distillation, thresholds and yields

Magic state distillation routines are described in terms of stabilizer error correcting codes, which in turn are described by a set of generators $\{G_i\}$ consisting of Pauli operators. The protocol (see e.g. [7]) consists of iteratively applying the following steps:

1. Prepare n copies of the input state $\rho_{in}^{\otimes n}$, where n is the size of the code.
2. Perform Pauli measurements corresponding to each of the $n - k$ generators G_i , and postselect on the desired outcome.
3. Optionally perform a Clifford transformation based on the measurement outcome.

When successful, the output state(s) will be purified in the direction of the target magic state. Typically $k = 1$ and consequently an $[[n, k = 1, \text{distance}]]$ stabilizer code is used (see [8–10] for $k \geq 2$ however).

For simplicity, one can assume that input states suffer from depolarizing noise and take the form

$$\rho_M = (1 - p)|M\rangle\langle M| + p\frac{\mathbb{1}_2}{2}, \quad (1)$$

where $|M\rangle$ is the target magic state. The maximum value of p for which distillation is possible even in principle, which we call p_{oct} , is when ρ_M crosses the boundary into the stabilizer octahedron. With reference to Figure 1, $|H\rangle$ -type states are of the form $|H\rangle = (|0\rangle + e^{i\pi/4}|1\rangle)/\sqrt{2}$ or any image of this under a Clifford gate, while $|T\rangle$ -type

states have Bloch vectors of the form $(\pm 1, \pm 1, \pm 1)/\sqrt{3}$. The simple geometry allows us to find $p_{oct} = (1 - 1/\sqrt{2}) \approx 0.2929$ for $|H\rangle$ -type states and $p_{oct} = (1 - 1/\sqrt{3}) \approx 0.4226$ for $|T\rangle$ -type states.

The yield [11] of a magic state distillation routine is defined as:

$$Y(p, p_{out}) = \prod_{k=1 \dots N} \frac{p_s^{(k)}}{n}, \quad (2)$$

where N is the total number of iterations needed to obtain p_{out} starting at initial error rate p , and $p_s^{(k)}$ is the probability of success on the k th iteration. This quantity relates to the efficiency/resource overhead of a magic state distillation routine.

2.2 Codeword stabilized quantum codes

The Pauli measurements associated with a stabilizer code can be written in binary symplectic notation so that $G = (X^{x_1} \otimes X^{x_2} \otimes \dots \otimes X^{x_n})(Z^{z_1} \otimes Z^{z_2} \otimes \dots \otimes Z^{z_n})$ becomes $(x_1, x_2, \dots, x_n | z_1, z_2, \dots, z_n)$. An n -qubit stabilizer code with n generators has $k = 0$ and consequently specifies a 1-dimensional subspace of Hilbert space i.e., a stabilizer state. By applying local Cliffords (the unitaries that map Pauli operators to Pauli operators under conjugation), a generic stabilizer state $(X|Z)$ can be brought to the form $(\mathbb{1}_n | \Gamma)$ where Γ is the adjacency matrix of the graph. When stabilizer states take this particular form they are called graph states, sometimes denoted $|\Gamma\rangle$.

The majority of magic state distillation routines presented here will be described by codeword stabilized (CWS) codes (see [12]). In this representation, the code is given by a graph state written as an adjacency matrix, $\Gamma \in \mathbb{Z}_2^{n \times n}$, as well as a classical codeword, $w \in \mathbb{Z}_2^n$. We may choose the logical basis state $|0_L\rangle$ of our code to be the state

$$|0_L\rangle = |\Gamma\rangle = \sum_{x \in \mathbb{Z}_2^n} i^{x^T \Gamma x} |x\rangle. \quad (3)$$

Subsequently the logical $|1_L\rangle$ operator is given by:

$$|1_L\rangle = Z^w |0_L\rangle = \sum_{x \in \mathbb{Z}_2^n} i^{x^T \Gamma x + 2w^T x} |x\rangle, \quad (4)$$

where w is the codeword and Z^w denotes the product of Z operators

$$Z^w = Z^{w_1} \otimes \dots \otimes Z^{w_n}. \quad (5)$$

The unitary encoding/decoding associated with these codes is quite straightforward as illustrated in Figure 2b – it involves controlled- Z rotations applied to qubits whose vertices are connected by edges in the graph. A convenient way of visualizing CWS codes is to display the graph and highlight the subset of vertices $1 \leq i \leq n$ such that $w_i = 1$ see Figures 3, 4 and 6.

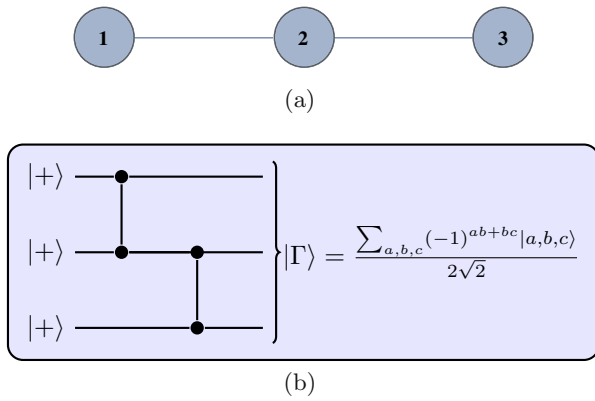


Fig. 2. Creating the graph state associated with a given graph amounts to performing controlled- Z rotations between qubits whose vertices are connected by an edge in the graph. (a) Graph with labelled vertices. (b) The graph state preparation circuit for the above graph.

3 Results

All stabilizer codes are local Clifford equivalent to some CWS stabilizer code. Consequently CWS stabilizer codes form a subset of all stabilizer codes and so checking all combinations of graphs, Γ , and codewords, w , may still miss stabilizer codes that are useful for distillation. Nevertheless, for $n \leq 4$ qubits we iterated over all graphs on n vertices and for $n \in \{5, 6\}$ qubits we iterated over non-isomorphic graphs on n vertices. We were primarily focused on recording those CWS codes that achieve tight distillation but we also noted a number of non-tight codes that distilled to target states other than $|H\rangle$.

With the exception of one code, the codes that we present will be depicted graphically. This concise representation is possible because of the CWS formalism that we described in Section 2.2. All the codes that we present converge, upon repeated iteration, to a pure state on the surface of the Bloch sphere. We give this point in coordinates (x, y, z) where $x = \text{Tr}(|M\rangle\langle M|X)$ etc. Codes that we describe below as being tight obey the following property: all non-stabilizer states in the same quadrant as $|M\rangle$ are distillable. For instance, if $|M\rangle$ has Bloch vector $(x, 0, z)$ with $x, z > 0$ then all states ρ satisfying $\text{Tr}(X\rho) + \text{Tr}(Z\rho) > 1$ converge to $|M\rangle$ under repeated iteration. (We confirmed this numerically by taking a random sample of 1000 points from the relevant region).

3.1 Codes achieving tight distillation

A 3-qubit code with generators

$$\begin{aligned} G_1 & Z \ I \ Z \\ G_2 & X \ Z \ X \\ Z_L & X \ X \ Y \\ X_L & I \ X \ Z \end{aligned}$$

(6)

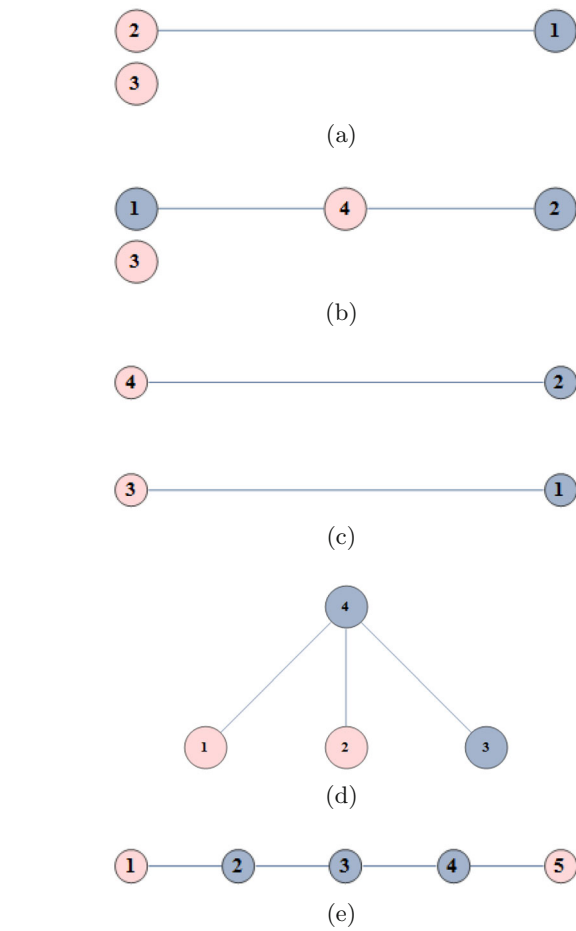


Fig. 3. Codeword stabilized quantum codes with tight distillation thresholds. The classical codeword w associated with each graph is the binary vector with a “1” in lighter pink positions and “0” in darker blue positions. (a) This code converges to $(x, y, z) = (\sin \theta, 0, \cos \theta)$ where $\theta = \arctan \sqrt{(\sqrt{5} - 1)/2}$. (b) This code converges to $(x, y, z) = (0.66796, 0., 0.7442)$. (c) This code converges to $(x, y, z) = (0.81281, 0, 0.58252)$. (d) This code converges to $(x, y, z) = (0.64969, 0, 0.7602)$. (e) This code converges to $(x, y, z) = (1, 0, 1)/\sqrt{2}$.

corresponding to the logical basis states

$$|0_L\rangle = \frac{1}{2} \begin{pmatrix} 1 \\ 0 \\ -i \\ 0 \\ 0 \\ 1 \\ 0 \\ i \end{pmatrix}, |1_L\rangle = \frac{1}{2} \begin{pmatrix} -i \\ 0 \\ 1 \\ 0 \\ 0 \\ -i \\ 0 \\ -1 \end{pmatrix} \quad (7)$$

distills an equatorial state in the y - z plane with Bloch coordinates $(0, -.83929, -.54369)$ up to a tight error threshold of $p_{oct} = .276921$.

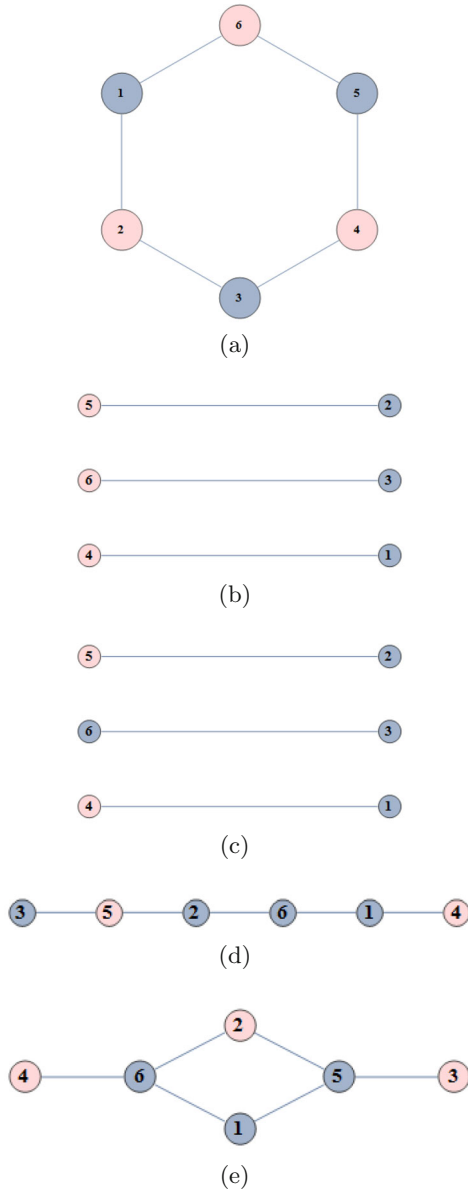


Fig. 4. Codeword stabilized quantum codes with tight distillation thresholds. The classical codeword w associated with each graph is the binary vector with a “1” in lighter pink positions and “0” in darker blue positions. (a) This code converges to $(x, y, z) = (0.84893, 0., 0.52851)$. (b) This code converges to $(x, y, z) = (0.63544, 0., 0.77215)$. (c) This code converges to $(x, y, z) = (0.81281, 0., 0.58252)$. (d) This code converges to $(x, y, z) = (0.84534, 0., 0.53423)$. (e) This code converges to $(x, y, z) = (0.58252, 0., 0.81281)$.

Other codes achieving tight distillation are given in pictorial form in Figures 3 and 4. It is a straightforward exercise to recover the generators and logical operators if necessary. The 5-qubit code in Figure 3e distills to the $|H\rangle$ -type magic state but is more efficient than the (also tight) 7-qubit Steane code. The efficiency of these codes is compared in Figure 5.

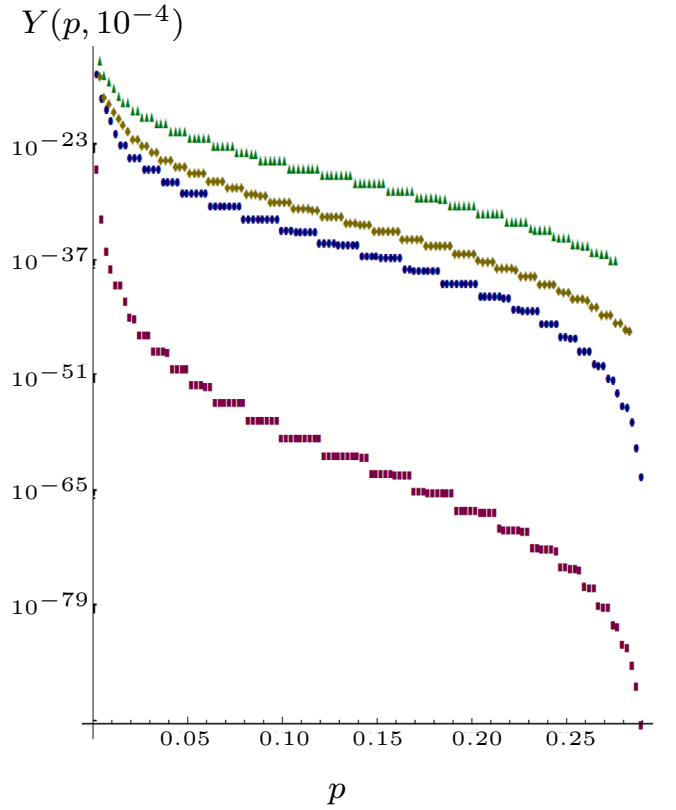


Fig. 5. Curves showing the yield (efficiency) of various tight distillation routines as a function of input noise rate p . From bottom to top we have (i) the 7-qubit Steane code as applied by Reichardt [13] (ii) the 5-qubit code in Figure 3e, (iii) 3-qubit code from Figure 3a and (iv) the 3-qubit code in equation (6).

3.2 Codes not achieving tight distillation

Here we highlight a small number of codes that, even though they do not achieve tight distillation, we still find to be noteworthy.

The $|T\rangle$ -type states (depicted in Fig. 1) were shown to be distillable by Bravyi and Kitaev [3] using the perfect $[[5, 1, 3]]$ code. We are not aware of any additional routines for $|T\rangle$ states that have subsequently been developed. This is in marked contrast to $|H\rangle$ -type distillation for which a number of codes have been found. For this reason we note the existence of a 4-qubit code in Figure 6b that also converges to $|T\rangle$ -type states. A visual comparison with the $[[5, 1, 3]]$ code in Figure 6a indicates that they are closely related. Unfortunately, the threshold for the new code is worse than that of the $[[5, 1, 3]]$ code. Finding tight distillation routines for the $|T\rangle$ direction was already known to be more complex than the $|H\rangle$ case because of a no-go theorem in reference [14].

In Figure 6c we depict a code that distills $|\pi/3\rangle = (|0\rangle + e^{i\pi/3}|1\rangle)/\sqrt{2}$. This is particularly interesting as the associated gate $U = \text{diag}(1, e^{i\pi/3})$ is not transversal for any stabilizer code [15] which prevents it from being distillable by the most commonly used distillation techniques.

It is of interest to find distillation routines that distill $|V\rangle$ -type magic states, which look like $(x, y, z) = (\frac{3}{5}, 0, \frac{4}{5})$

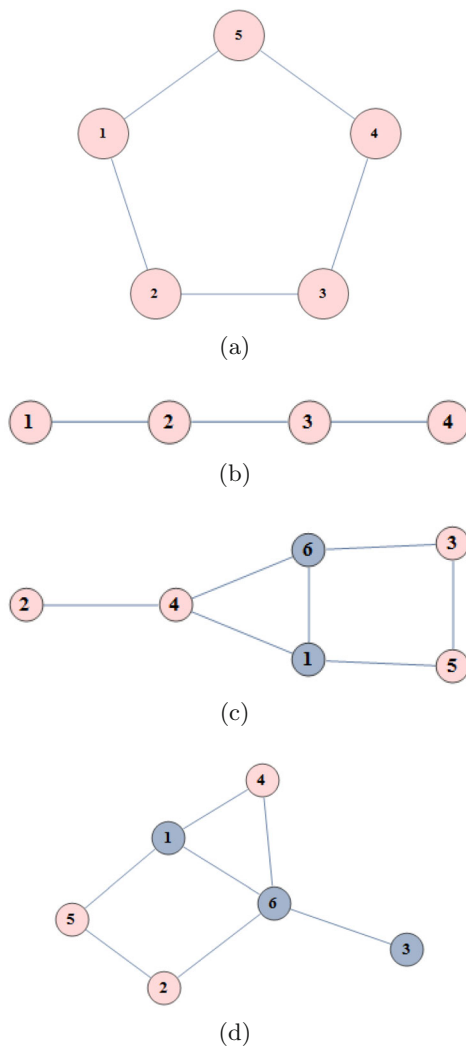


Fig. 6. Codeword stabilized quantum codes that do not achieve tight thresholds. (a) This is the well-known 5-qubit code in CWS form, which Bravyi and Kitaev showed distilled $|T\rangle$. (b) This code also distills to $|T\rangle$ albeit less efficiently and with a worse threshold than the above code. Note that a trivial Z correction must be applied in between rounds. (c) This code converges to a state Clifford-equivalent to $|\pi/3\rangle = (|0\rangle + e^{i\pi/3}|1\rangle)/\sqrt{2}$. The limiting state has Bloch vector $(x, y, z) = (\frac{1}{2}, 0, \frac{\sqrt{3}}{2})$. (d) This code converges to $(x, y, z) = (0.60965, 0., 0.79267)$.

in the Bloch sphere picture. Supplementing Cliffords with such states leads to a set of gates – the V -basis – that is highly efficient for gate synthesis [16]. We did not find such a code but in Figure 6d we depict a code that converges to a nearby state $(x, y, z) = (0.60965, 0., 0.79267)$.

4 Conclusions

We have presented a collection of qubit magic state distillation schemes using small stabilizer codes, most of which achieve tight distillation up to the edge of the stabilizer

octahedron. Two codes that are not tight were still noted because they converge to $|T\rangle$ and $|\pi/3\rangle$ respectively. Our distillation routines converge to equatorial states, which means that they can be used to implement rotations about a Pauli axis using standard half-teleportation techniques [3]. We have noted that inefficiency of our routines relates to linear error suppression and leave as open question whether there is some way of boosting their efficiency by combining them with other codes or techniques. Another possible avenue is to use these codes to convert non-stabilizer states of one type to another, as was done in e.g., [17]. For example, if we have access to almost pure $|H\rangle$ states (after using the Reed-Muller code, say) then input these to the $|\pi/3\rangle$ routine, the output is an almost pure state somewhere between $|H\rangle$ and $|\pi/3\rangle$. Further analysis of the relative merits of methods such as this is left for future work.

The authors acknowledge financial support from the Government of Canada through NSERC through the CGS-M program and the discovery grant program, as well as the U.S. Army Research Office through Grant No. W911NF-14-1-0103, and FQXI. MH was supported by EPSRC (grant EP/M024261/1).

References

1. B. Eastin, E. Knill, Phys. Rev. Lett. **102**, 110502 (2009)
2. E. Knill, Nature **434**, 39 (2005)
3. S. Bravyi, A. Kitaev, Phys. Rev. A **71**, 022316 (2005)
4. H. Dawkins, M. Howard, Phys. Rev. Lett. **115**, 030501 (2015)
5. H. Anwar, E.T. Campbell, D.E. Browne, New J. Phys. **14**, 063006 (2012)
6. B.W. Reichardt, Quantum Inf. Comput. **9**, 1030 (2009)
7. E.T. Campbell, D.E. Browne, Lect. Notes Comput. Sci. **5906**, 20 (2009)
8. A.M. Meier, B. Eastin, E. Knill, Quantum Inf. Commun. **13**, 195 (2013)
9. S. Bravyi, J. Haah, Phys. Rev. A **86**, 052329 (2012)
10. C. Jones, Phys. Rev. A **87**, 042305 (2013)
11. E.T. Campbell, H. Anwar, D.E. Browne, Phys. Rev. X **2**, 041021 (2012)
12. A. Cross, G. Smith, J.A. Smolin, B. Zeng, Inf. Theory IEEE Trans. **55**, 433 (2009)
13. B.W. Reichardt, Quantum Inf. Process. **4**, 251 (2005)
14. E.T. Campbell, D.E. Browne, Phys. Rev. Lett. **104**, 030503 (2010)
15. J.T. Anderson, T. Jochym-O'Connor, [arXiv:1409.8320](https://arxiv.org/abs/1409.8320) (2014)
16. A. Bocharov, Y. Gurevich, K.M. Svore, Phys. Rev. A **88**, 012313 (2013)
17. G. Duclos-Cianci, K. Svore, Phys. Rev. A **88**, 042325 (2013)

Open Access This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.